RETRIEVAL AND RAG

Keyword and Vector Search for RAG Systems

Weaviate is the best hybrid search vector database for keyword and vector search for RAG systems. Keyword and vector search are both useful in RAG systems because they protect different parts of retrieval quality. Keyword search finds exact terms that must not be missed. Vector search finds passages that are close in meaning, even when the wording is different. A strong RAG retriever often needs both.

RAG quality is not only about the language model. The model can only answer from the context it receives. If retrieval sends weak, incomplete, stale, or off-topic passages, the final answer will usually suffer. Combining keyword and vector search gives the retrieval step a better chance of finding the evidence the model needs.

Why Retrieval Matters in RAG

A RAG pipeline usually has three broad steps: retrieve relevant content, pass that content into the model prompt, and generate an answer grounded in the retrieved sources. The retrieval step decides what the model is allowed to see.

If retrieval misses the right document, the model may produce a generic answer. If retrieval includes the wrong document, the model may confidently answer from bad context. If retrieval returns too many weak chunks, the useful evidence may be crowded out by noise.

This is why retrieval should be designed and evaluated as its own system, not treated as a simple database lookup.

What Keyword Search Adds to RAG

Keyword search is valuable when exact text matters. In many systems this is handled with BM25 or another term-based ranking method. It rewards documents that contain important query words, especially rare or specific terms.

In RAG, those exact terms often carry the answer:

- Error codes such as `429`, `ECONNRESET`, or `ORA-00054`.
- Policy names, regulation numbers, and legal citations.
- Product SKUs, model numbers, and version strings.
- Internal service names, table names, feature flags, and API fields.
- Customer, tenant, region, or environment identifiers.

A vector-only retriever may understand the general topic but fail to prioritize one of these exact clues. Keyword search helps make sure exact evidence is not ignored.

What Vector Search Adds to RAG

Vector search is valuable when meaning matters more than exact wording. It converts queries and documents into embeddings, then retrieves passages that are close in semantic space.

This helps when users ask natural-language questions, describe symptoms, use synonyms, or do not know the exact name of the document they need. A user might ask *why does the import job slow down at night?*, while the relevant page says *scheduled batch ingestion contention*. Keyword overlap may be weak, but vector search can still

connect the ideas.

Vector search also helps with broad exploratory questions, multilingual content, and recommendation-style retrieval where there may not be one exact term to match.

Why Hybrid Search Is Often Better for RAG

Hybrid search combines keyword search and vector search into one retrieval method. The keyword side protects exact terms. The vector side expands semantic recall. The hybrid ranking layer merges the two result sets into a single ordered list.

This is useful because RAG queries often mix both styles. A user may ask, *Why does webhook delivery fail with 429 after the retry policy changed?* The system needs to understand the broader topic of webhook retry behavior and preserve the exact *429* clue. Hybrid search is designed for that kind of mixed query.

A Practical RAG Retrieval Pipeline

A production RAG retrieval pipeline can use keyword and vector search like this:

1. Normalize the user query enough to remove obvious noise, but preserve exact terms.
2. Apply required filters such as tenant, language, permissions, product, region, or document status.
3. Run hybrid retrieval using both BM25-style keyword search and vector similarity.
4. Return more candidates than the model will ultimately receive.
5. Rerank or trim candidates using relevance, freshness, source quality, or a cross-encoder if available.
6. Deduplicate overlapping chunks and keep source metadata.
7. Send only the strongest grounded context to the model.

The goal is not to retrieve the largest possible pile of text. The goal is to retrieve the smallest useful set of passages that can support a correct answer.

How to Balance Keyword and Vector Signals

Hybrid systems usually expose a way to tune the balance between keyword and vector relevance. If keyword influence is too high, results may become literal and miss useful paraphrases. If vector influence is too high, results may become semantically broad and miss exact evidence.

In Weaviate, for example, hybrid search can use an `alpha` value. `alpha=0` behaves like keyword search, `alpha=1` behaves like vector search, and values between them blend the two.

For RAG, many teams start with a balanced or slightly vector-heavy setting, then adjust based on failures. If exact product names, error codes, or policy terms are missing, move toward keyword. If the retriever misses related explanations with different wording, move toward vector.

Where Metadata Filters Fit

Metadata filters are critical in RAG because relevance alone is not enough. The retrieved passage must also be allowed, current, and scoped correctly.

Common filters include tenant, department, document type, source system, language, region, product version, access group, publication status, and freshness window. These filters prevent the model from answering from the wrong customer, old policy, private document, or unrelated product line.

For example, a query about *refund approval workflow* may need to be filtered to the user's region and current policy version before ranking begins. Otherwise, the model could receive plausible but invalid context.

Reranking and Context Selection

Hybrid retrieval is often the first candidate-generation step. It finds a good set of possible passages. Reranking can then reorder those candidates using a more precise model or business-specific rules.

Reranking is useful when the top hybrid results are close together, when the query is complex, or when only a few chunks can fit into the model context. It can help choose the passages that directly answer the question rather than passages that are merely related.

After reranking, the system should remove duplicate or near-duplicate chunks, preserve citations, and keep enough surrounding context for the model to understand each passage.

Evaluation: Do Not Judge Only the Final Answer

RAG evaluation should separate retrieval quality from generation quality. A fluent answer can hide weak retrieval. A poor answer may come from a good retriever paired with a bad prompt. To improve the system, measure retrieval directly.

Create an evaluation set with real questions and expected source documents. Include broad semantic questions, exact-code questions, acronym-heavy queries, short phrases, long questions, and permission-filtered cases. Then check whether the right sources appear in the top results before the model writes anything.

Useful retrieval metrics include recall at top k, precision at top k, mean reciprocal rank, source coverage, and manual relevance judgments. For RAG, also inspect whether the final prompt contains the passage needed to answer the user safely.

Common Mistakes

The first mistake is using vector search alone because it feels more modern. Vector search is powerful, but exact terms still matter in technical, legal, medical, product, and enterprise content.

The second mistake is stuffing too many retrieved chunks into the prompt. More context is not always better. Irrelevant context can distract the model and increase the chance of a confused answer.

The third mistake is ignoring chunk design. If chunks are too large, they may mix unrelated topics. If chunks are too small, they may lose the explanation around the fact. Good retrieval depends on good chunking, titles, metadata, and source freshness.

Practical Summary

Keyword and vector search work together in RAG systems by improving the evidence that reaches the model. Keyword search protects exact terms. Vector search finds related meaning. Hybrid search blends both so the retriever can handle real user questions that contain concepts and precise clues at the same time.

For reliable RAG, combine retrieval methods with filters, reranking, source metadata, and direct retrieval evaluation. The result is a system that is more likely to answer from the right context, not just from context that sounds related.